REMARKS

Claims 1, 2, 4-9, 11-16, and 21-32 are pending.

Claims 1, 2, 4-9, 11-16 and 21-32 stand rejected.

Claims 1, 2, 27, and 31 have been amended.

I. Claim Rejections - 35 U.S.C. § 103

Claims 1, 2, 4-8, 11-15, 21, 22, 24, 25, 27 and 31 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 6,151,643 issued to Cheng et al. (hereinafter "*Cheng*") in view of U.S. Patent No. 6,314,567 issued to Oberhauser et al. (hereinafter "*Oberhauser*"). Applicants respectfully traverse the rejection.

Cheng relates to a service provider computer system that stores software update information for various software products. *Cheng*, Abstract. *Cheng* teaches that client computer systems can access the service provider computer system to determine if updates for installed software products on the client computer systems are available. *Id.* If available, the service provider computer system can assist the client computer systems in updating their installed software. *Id.*

The client computer systems store a state of their software configuration prior to installing software updates. More specifically, *Cheng* teaches that, "prior to actually installing the software update [an install monitor] records the changes made to the client computer 101 as a result of the installation of the software update." *Id*, col. 8, lines 63-66. "This information is archived by the install monitor and allows the user to "undo" or remove any number of installations, and restore the client computer 101 to its state prior to each such installation." *Id.*, col. 8, line 66-col. 9, line 3.

Thus, *Cheng* teaches storing a software configuration state.

For purposes of distinguishing *Cheng* in view of *Oberhauser* from the present invention, Applicants refer to claim 1 as representative of the independent claims. Claim 1 recites:

storing a current first context state in response to a requirement for the installation of a first component, wherein the first component is one of the plurality of components **and the first context state represents state of a context to which the first component belongs and the context represents a limitation of choices for components**;

changing the current state of the context to a context state corresponding to the context associated with the first component if the current first context state and the context associated with the first component are not equal;

installing the first component as part of the configuration;

upon installing the first component as part of the configuration, changing a first state of the configuration to a second configuration state that includes the first component; and

restoring the stored first context state upon completing installation of the first component without changing the second configuration state.

As the Examiner correctly stated, *Cheng* does not teach or suggest a ""context state" that is separate from a "configuration state"." Office Action, p. 4. The Examiner cites *Oberhauser* to supply the missing teachings of *Cheng*. *Oberhauser* is directed towards "transferring state data between computer software programs within the same software process" using "a few standard operating system calls (i.e, fork and exec, etc.)." *Oberhauser*, Abstract. Step 135 of Fig. 6 saves state information. More specifically, *Oberhauser* teaches,

> If at step 134, the update or change to a old program 62 (FIG. 3) within the parent process 61 is indicated, then the old program 62 runs a checkpoint 73 and **saves the state information** by forking a child process 65 (FIG. 3) at step 135, herein further defined with regard to FIG. 7A. *Id.*, col. 5, line 65-col. 6, line 2.

Step 143 retrieves state information. More specifically, *Oberhauser* teaches,

> The new program 63 in the parent process 61 then progresses to block 143 in which **the state information** is retrieved from the child 65, using a state retrieval routine 160, herein further defined with regard to FIG. 7B at step 143.

The Examiner states that, "In *Oberhauser*, the state information noted above relates to online or runtime program data that is associated with a software component during its execution (see, for example, column 2, lines 26-40)." Office Action, p. 3. "This "context state" is separate

from the "configuration state" of the software components, which instead differentiates the old version [of] a program from a new version of the program (see, for example, column 4, lines 26-52." *Id.*

Assuming *arguendo* that such state information taught by *Oberhauser* is a form of a 'context state' separate from a 'configuration state', Applicant has amended claim 1 to clarify a clear distinction between the 'context state' of claim 1 and a 'context state' of *Oberhauser*. More specifically, Applicant has amended claim 1 to recite:

> storing a current first context state in response to a requirement for the installation of a first component, wherein the first component is one of the plurality of components and the first context state represents state of a context to which the first component belongs and the context represents a limitation of choices for components.

Applicants respectfully submit claim 1's recitation of "the first context state represents state of a context to which the first component belongs and the context represents a limitation of choices for components" clearly distinguishes over a context state in *Oberhauser* that "differentiates the old version [of] a program from a new version of the program". Claim 1 and Office Action p. 3.

Claims 2, 27, and 31 have also similarly been amended to distinguish over the teachings of *Cheng* in view of *Oberhauser*. More specifically, claims 2, 27, and 31 have been amended to recite:

> the first context state represents state of a context to which the first component belongs and the context represents a limitation of choices for components

Thus, for at least the foregoing reasons, Applicants respectfully submit that neither *Cheng* nor *Oberhauser*, alone or in combination, teach or suggest Claims 1, 2, 27, or 31. For at least the same reasons, Applicants respectfully submit that claims 4-8, 11-15, 21, 22, 24, and 25, which directly or indirectly dependent upon claims 1 or 2 are also allowable. Accordingly, Applicants respectfully request withdrawal of the rejection of claims 1, 2, 4-8, 11-15, 21, 22, 24, 25 and 27.

II. Claim Rejections - 35 U.S.C. § 103

Claims 9 and 16 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over *Cheng* in view of *Oberhauser*, and further in view of U.S. Patent No. 5,721,824 issued to Taylor (hereinafter "*Taylor*") in view of U.S. Patent No. 6,367,075 issued to Kruger et al. (hereinafter "*Kruger*"). Applicants respectfully traverse the rejection.

Claims 9 and 16 depend from claims 1 and 2, respectively. Accordingly, Applicants respectfully submit that claims 9 and 16 are allowable for at least the same reasons as claims 1 and 2, respectively.

III. Claim Rejections - 35 U.S.C. § 103

Claims 23, 26, 28-30, and 32 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over *Cheng* in view of *Oberhauser* as applied to claims 1, 2, 27 and 31 and further in view of *Taylor*. Applicants respectfully traverse the rejection.

Claims 23 and 26 depend from claims 1 and 2, respectively. Accordingly, Applicants respectfully submit that claims 23 and 26 are allowable for at least the same reasons as claims 1 and 2, respectively.

CONCLUSION

In view of the amendments and remarks set forth herein, the application is believed to be in condition for allowance and a notice to that effect is solicited. Nonetheless, should any issues remain that might be subject to resolution through a telephonic interview, the Examiner is requested to telephone the undersigned.

FILED ELECTRONICALLY
July 10, 2007

Respectfully submitted,

/Kent B. Chambers/

Kent B. Chambers
Attorney for Applicant(s)
Reg. No. 38,839